



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



TESTOVÁNÍ DPS POMOCÍ PLOCHÉHO SCANNERU

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Patrik Odcházal**
Vedoucí práce: prof. Ing. Zdeněk Plíva, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

TESTING PCB USING A FLATBED SCANNER

Bachelor thesis

Study programme: B2646 – Information technology
Study branch: 1802R007 – Information technology
Author: **Patrik Odcházal**
Supervisor: prof. Ing. Zdeněk Plíva, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Patrik Odcházal**
Osobní číslo: **M10000179**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Testování DPS pomocí plochého scanneru**
Zadávající katedra: **Ústav informačních technologií a elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s algoritmy zpracování obrazu a s vhodným vývojovým prostředím.
2. Navrhněte aplikaci pro zpracování a vyhodnocení obrazových dat.
3. Vyzkoušejte možnosti metody na dodaných vzorcích a vyhodnoťte její možnosti.

Rozsah grafických prací:

Dle potřeby dokumentace

Rozsah pracovní zprávy:

cca 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Hlaváč V., Sedláček M.: Zpracování signálu a obrazu, Skripta FEL ČVUT, Praha, 1999
- [2] Bass M.: Handbook of Optics, vol. II, ISBN-10: 0071498907
- [3] Tutorial NI Vision Assistant (online na <http://physics.ucsd.edu/neurophysics/Manuals/NationalInstruments/NIVisionAssistantTutorial.pdf>)

Vedoucí bakalářské práce:

prof. Ing. Zdeněk Plíva, Ph.D.

Ústav informačních technologií a elektroniky

Konzultant bakalářské práce:

Ing. Jiří Horčíčka

Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce:

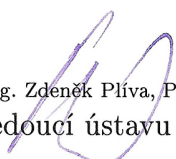
12. září 2013

Termín odevzdání bakalářské práce:

16. května 2014


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Ing. Zdeněk Plíva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2013

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Datum:

Podpis:

Poděkování

Chtěl bych poděkovat všem, kteří jakkoliv přispěli při zpracování mé bakalářské práce. V první řadě mé největší poděkování patří prof. Ing. Zdeňku Plívovi, Ph.D., vedoucímu bakalářské práce, a také Ing. Jiřímu Horčíčkovi, konzultantovi bakalářské práce, kteří mi byli po celou dobu nápomocni, za jejich ochotu a trpělivost, cenné rady a připomínky při psaní této práce.

Dále bych chtěl poděkovat mé rodině a přátelům za podporu během studia.

Abstrakt

Bakalářská práce se zabývá vytvořením aplikace pro testování desky plošných spojů pomocí plochého scanneru. Výstupem aplikace je obrázek vyrobené desky plošných spojů, na kterém jsou barevně označeny místa, která se neshodují s obrázkem předlohy. Díky tomu pomůže aplikace uživateli rozhodnout, zda je deska plošných spojů vyrobena správně a zda nemá vady elektrických spojů.

Práce zahrnuje krátké seznámení s metodami pro testování desek plošných spojů. Následuje výčet několika možností řešení zadaného problému pomocí různých programových nástrojů. Z nich jsou vybrány aplikace National Instruments Vision Assistant (návrh skriptu zpracovávajícího vstupní data) a vývojové prostředí LabVIEW (vytvoření uživatelského rozhraní a optimalizaci skriptu). Důležitou součástí je samotná aplikace uložená na přiloženém CD. Popis funkcí a prostředí aplikace je uveden v textu této práce.

Klíčová slova

LabVIEW, National Instruments Vision Assistant, OpenCV, Qt Creator, deska plošných spojů, plochý scanner, Geometric Matching, Golden Template Comparison, metody pro testování DPS

Abstract

The bachelor thesis deals with creating application for testing printed circuit boards using a flatbed scanner. The output of this application is image of produced PCB with color highlight defects. The application helps the user to decide whether the PCB is made correctly and that doesn't have electrical connections defects.

The thesis includes a short introduction to the methods for testing printed circuit boards. The following is a list of possible solution to solve this problem using various software tools. Selected applications were National Instruments Vision Assistant (the script for processing the input) and LabVIEW (creation of user interfaces and optimization the script).

An important part is the application stored on the supplied CD. Description of the functions and the application environment is presented in the text of this thesis.

Key words

LabVIEW, National Instruments Vision Assistant, OpenCV, Qt Creator, Printed Circuit Board, flatbed scanner, Geometric Matching, Golden Template Comparison, PCB inspection technique

Obsah

Úvod	12
1 Základní pojmy	13
1.1 Desky plošných spojů	13
1.2 Plochý scanner	13
2 Grafické formáty	14
2.1 BMP (Bitmap File Format)	14
2.2 PNG (Portable Network Graphics)	14
2.3 TIFF (Tagged Image File Format)	15
3 Vstupní data	16
4 Metody pro testování DPS	17
4.1 Neelektrické metody	17
4.1.1 Vizuální kontrola	17
4.1.2 Automatická optická inspekce	17
4.2 Elektrické metody	18
5 Možnosti řešení	19
5.1 Programovací jazyk C++ a knihovna OpenCV	19
5.2 Qt Creator	19
5.3 National Instruments Vision Assistant	20
5.4 Vývojové prostředí National Instruments LabVIEW	20
5.4.1 Čelní panel	21
5.4.2 Blokový diagram	21
6 Návrh řešení	22
6.1 Skript pro zpracování obrazu	22
6.2 Geometric Matching	23
6.2.1 Hledání křivek v obraze	24
7 Aplikace pro testování DPS	27
7.1 Uživatelské rozhraní	27
7.2 Vytvoření šablon	28
7.2.1 Šablona pro Geometric Matching	28

7.2.2	Šablona pro Golden Template Comparison	29
7.3	Inspekce	30
7.3.1	Příprava pro porovnání	31
7.3.2	Hledání motivu	31
7.3.3	Porovnání motivu	32
7.3.4	Zobrazení defektů	32
7.3.5	Uložení výsledku	33
8	Vyhodnocení aplikace	35
8.1	Metoda Feature-based	36
8.2	Metoda Edge-based	36
	Závěr	37
	Literatura	38
	Přílohy	40
A	Obsah přiloženého CD	40

Seznam obrázků

3.1	Obrázek předlohy	16
3.2	Naskenovaná DPS	16
4.1	AOI tester Orbotech Fusion 22	18
4.2	Elektrický tester ATG A5 Neo	18
6.1	Geometrické tvary	24
6.2	Hledání křivek	25
7.1	Uživatelské rozhraní aplikace	28
7.2	Výběr oblasti pro porovnání	29
7.3	Vytvoření šablony	30
7.4	Schéma Inspection.vi	31
7.5	Schéma ShowDefects.vi	33
7.6	Výsledek inspekce	34
8.1	Detail posunutí vyvrtaného otvoru	35

Seznam pojmů a zkratk

A/D	analogově digitální převodník
AOI	automatická optická inspekce
BMP	Bitmap File Format, je grafický formát
CCD	Charge-coupled Device, je senzor pro snímání obrazové informace
Cluster	seskupení
CMYK	barevný model se subtraktivním mícháním barev
DPS	deska plošných spojů
Embedded	jednouúčelový počítačový systém
IPTC	standard pro ukládání textových informací do obrazových souborů
LabVIEW	Laboratory Virtual Instruments Engineering Workbench, je laboratorní pracoviště virtuálních nástrojů
LZW	Lempel-Ziv-Welch, je bezztrátový kompresní algoritmus
MATLAB	Matrix Laboratory, je programové prostředí a skriptovací programovací jazyk
NI	National Instruments
OpenCV	Open Source Computer Vision Library, je knihovna s algoritmy a funkcemi pro zpracování obrazu
PCB	Printed Circuit Board
PNG	Portable Network Graphics, je grafický formát
RGB	barevný model s aditivním mícháním barev
RLE	Run-length Encoding, je bezztrátový kompresní algoritmus
SubVI	podprogram ve vývojovém prostředí LabVIEW
TIFF	Tagged Image File Format, je grafický formát
TUL	Technická univerzita v Liberci
VI	Virtual Instrument, je virtuální přístroj
ZIP	bezeztrátový kompresní algoritmus

Úvod

Elektronika se stala nedílnou součástí našeho života. Každý den se setkáváme s několika elektronickými zařízeními, které nám poskytují své funkce. Desky plošných spojů se používají ve všech elektronických zařízeních, a proto je zapotřebí tyto součástky vyrábět s co nejnižšími náklady. Z tohoto důvodu se desky plošných spojů vyrábějí sériově a mohou tak vzniknout vadné kusy. Proto je důležitá jejich kontrola ještě před osázením elektronických součástek.

Cílem bakalářské práce je vytvoření aplikace, která uživateli pomůže rozhodnout, zda se vyrobená deska plošných spojů shoduje s předlohou, či nikoliv. Pro řešení tohoto problému je důležité vybrat vhodné vývojové prostředí a seznámit se s algoritmy pro zpracování obrazu.

K vyhodnocení bude aplikace používat obrázek předlohy, který poskytne software pro návrh plošných spojů a obrázek hotového plošného spoje získaný z plochého scanneru. Obrázek předlohy bude vyžadován ve formátu *PNG* nebo *BMP* a naskenovaný motiv ve formátu *TIFF*. Na základě těchto zdrojových obrázků aplikace vyhodnotí rozdíly a graficky je zobrazí na předloze. Uživatelské rozhraní aplikace umožní uživateli nahrání zdrojových obrázků a jejich případné zrcadlení.

Aplikace bude otestována na několika vzorcích a ze získaných výsledků budou vyhodnoceny její možnosti.

1. Základní pojmy

1.1 Desky plošných spojů

Desky plošných spojů (DPS) zajišťují upevnění a vodivé propojení součástek v elektronických zařízeních. Jako základ desek se většinou používá skelný laminát, na který je nanesena vodivá vrstva - měděná fólie. Při výrobě spojového obrazce je nutné měděnou fólii vhodným způsobem zakrýt v místech, kde ji chceme zachovat. K tomu slouží tzv. vykrývací maska, pomocí které na desce vytvoříme požadovaný spojový obrazec. Následně je deska ponořena do leptací lázně, čímž dojde k odleptání nepotřebné mědi. [11]

K dispozici je celá řada programů, které umožňují rychlý a jednoduchý návrh plošného spoje. Přenos předlohy na měděnou fólii je možné provést několika způsoby. První způsob využívá plotteru, který nanáší krycí barvu a tím vytváří vykrývací masku. Dalším způsobem je tzv. fotocesta, která spočívá v použití desky, a ta má na měděné fólii nanesenou fotocitlivou vrstvu. Na tuto vrstvu se přiloží předloha vytištěná na průhlednou fólii a poté se nechá osvětit silným zdrojem světla. Následuje vyvolání desky v roztoku hydroxidu sodného. V obou těchto případech musíme desku nechat vyleptat. K leptání se používá například roztok chloridu železitého nebo roztok kyseliny chlorovodíkové a peroxidu vodíku. Posledním způsobem pro výrobu desky plošných spojů je gravírování spojového obrazce do měděné fólie pomocí laseru. [8]

1.2 Plochý scanner

Plochý scanner je vstupní zařízení sloužící k převodu tištěného textu a grafiky do elektronické podoby. Snímaná předloha se vkládá na skleněnou plochu scanneru a přiklopí se víkem. Pod sklem je umístěný světelný zdroj, který se pohybuje a osvětluje předlohu. Světlo odražené od předlohy se odráží přes soustavu zrcadel na CCD snímač. Čím je osvětlená plocha tmavší, tím méně světla se od ní odráží. CCD snímač převádí světlo na elektrické impulsy. Ty jsou převedeny do digitální podoby pomocí A/D převodníku. Výstup z převodníku zpracovává grafický čip a poté je přenesen do počítače.

2. Grafické formáty

V této kapitole se seznámíme s grafickými formáty použitými pro obrázek předlohy a pro obrázek vyrobené DPS získaný z plochého scanneru.

2.1 BMP (Bitmap File Format)

Formát BMP se používá v operačních systémech Microsoft Windows pro ukládání rastrové grafiky. BMP podporuje obrázky s barevnou hloubkou 1, 4, 8, 16, 24 a 32 bitů na pixel. Barevná hloubka určuje počet barev, kterých může nabývat každý pixel obrázku. Pro 4 a 8 bitů na pixel mohou být obrazová data uložena s kompresí RLE. U tohoto grafického formátu se komprese většinou nepoužívá. [7, s. 23]

Počet barev v závislosti na počtu bitů na pixel:

- 1 bit na pixel - dvě barvy;
- 4 bity na pixel - 16 barev;
- 8 bitů na pixel - 256 barev;
- 24 bitů na pixel - 16 mil. barev (tzv. TrueColor);
- 32 bitů na pixel - 4 mld. barev.

2.2 PNG (Portable Network Graphics)

Formát PNG byl vyvinut jako náhrada formátu GIF, který měl patentovaný kompresní algoritmus LZW a jeho použití bylo zpoplatněné. PNG nabízí barevnou hloubku až 48 bitů na pixel pro plnohodnotné uložení barev. Formát podporuje 8 nebo 16bitový alfa kanál, díky kterému je možné nastavit části obrázku různě průhledné.

Formát používá dvoustupňovou bezztrátovou metodu komprese. V prvním kroku mohou být použity filtry, které mají za úkol připravit data pro optimální kompresi. Existují čtyři základní typy filtrů, které jsou aplikovány na jednotlivé bajty, nikoliv na obrazové body. Existuje také možnost filtry nepoužít, v tomto případě jsou bajty nezměněny. V druhém kroku je na data aplikována kompresní metoda DEFLATE, která používá kombinaci slovníkového algoritmu LZ77 a Huffmanova kódování. [12, s. 379-380]

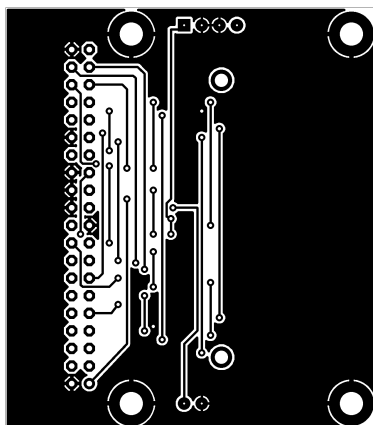
2.3 TIFF (Tagged Image File Format)

Dalším formátem pro bezztrátové ukládání rastrové grafiky je TIFF. Tento formát podporuje stejně jako PNG barevnou hloubku až 48 bitů na pixel. Na data může být aplikováno několik kompresních algoritmů, jako například LZW, RLE, ZIP a mnoho dalších. Je zde také možnost uložit data bez komprese.

Formát podporuje barevné modely RGB i CMYK a také například ukládání vrstev či více bitmap do jednoho souboru. Oproti PNG je možné do souboru TIFF uložit také tzv. IPTC nebo-li textové informace, které mohou obsahovat jméno autora, popis, klíčová slova a další. TIFF nabízí stejně jako PNG alfa kanál pro nastavení průhlednosti. [6, s. 40]

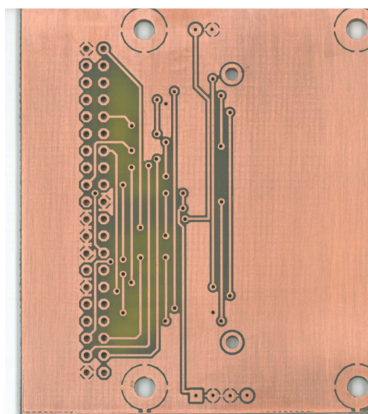
3. Vstupní data

Pro co nejpřesnější vyhodnocení grafické informace je nezbytné použít pro vstupní soubory grafické formáty, které jsou nekomprimované a nebo podporují bezztrátovou kompresi. Proto byl zvolen formát *PNG* nebo *BMP* pro obrázek předlohy a formát *TIFF* pro naskenovaný motiv. Obrázek 3.1 znázorňuje, jak může vypadat obrázek předlohy poskytnutý softwarem pro návrh plošných spojů.



Obrázek 3.1: Obrázek předlohy

Plochý scanner, který podporuje výstupní formát TIFF, poskytne obraz vyrobené DPS. Příklad výstupu ze scanneru znázorňuje obrázek 3.2.



Obrázek 3.2: Naskenovaná DPS

4. Metody pro testování DPS

Metody se dělí na dvě základní skupiny - elektrické a neelektrické. Obě skupiny mají své výhody i nevýhody a vzájemně se doplňují. Proto je na trhu k dostání několik typů přístrojů pro testování DPS. Testery se dělí do tří skupin - elektrické, optické a rentgenové.

4.1 Neelektrické metody

Jedná se o metody, které pro kontrolu DPS nepoužívají proměřování jejich elektrických obvodů. Patří sem vizuální kontrola a automatická optická kontrola.

4.1.1 Vizuální kontrola

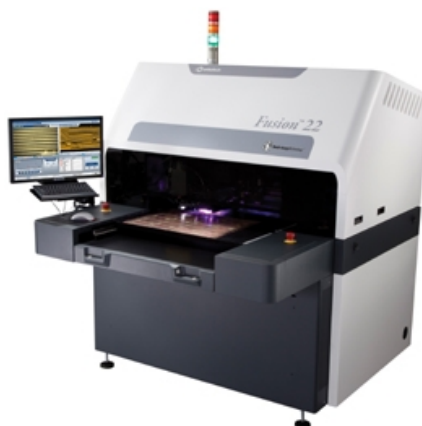
Vizuální kontrola je první z neelektrických metod, kterou provádí člověk. Musí být jasně určené, jaká odlišnost vyrobeného motivu je akceptovatelná a jaká už ne. Tento způsob kontroly je možný spíše u jednoduších motivů DPS, kde člověk postupným tréninkem získá zkušenosti a dokáže rychle posoudit, zda je výrobek použitelný nebo ne.

4.1.2 Automatická optická inspekce

Jedná se o bezkontaktní způsob kontroly vyrobené DPS, při kterém se výrobek nasnímká pomocí kamer a následně se porovnává se vzorem. Nalezené odchylky mohou, ale také nemusí mít vliv na funkčnost DPS. Systém proto nedokáže sám určit, zda se jedná o zmetek a nebo o funkční výrobek. Proto je nejlepší, pokud systém výsledek poskytne uživateli, a ten sám rozhodne, jestli je výrobek použitelný nebo ne. Tento způsob kontroly je možné použít pro všechny fáze výrobního procesu:

- kontrola neosázené DPS;
- kontrola osázených součástek;
- kontrola správného zapájení součástek.

Automatickou optickou inspekcí lze odhalit vady DPS, které by elektrické testování neodhalilo. Například pokud by byl spoj v nějakém místě extrémně zúžený, elektrická kontrola by tento problém nedetekovala. Proto je nejvhodnější kombinace několika druhů testování.



Obrázek 4.1: AOI tester Orbotech Fusion 22

Zdroj [1]

4.2 Elektrické metody

Elektrické testování patří mezi závěrečné zkoušky DPS, před jejím dalším zpracováním nebo odesláním zákazníkovi. Elektrické testery simulují na DPS funkci jejich vodičů a izolantů. Přístroj je vybaven několika sondami, které elektrické spoje proměřují a kontrolují, zda nedochází ke zkratu či přerušení vodiče.

Nejprve je prováděn test kontinuity, při kterém se zjišťuje, zda je mezi dvěma body očekávané elektrické propojení. Odpor, který je zjištěn při každém měření je porovnán s prahovou hodnotou a pokud je naměřená hodnota větší, je generováno chybové hlášení.

Poté se provádí test izolací. Test vyhodnocuje, zda je mezi sítěmi, které nemají být propojeny, dostatečná elektrická izolace. Naměřená hodnota by měla být větší než prahová, jinak to znamená, že je DPS špatně vyrobená a je opět generováno chybové hlášení. [5, s. 851-857]



Obrázek 4.2: Elektrický tester ATG A5 Neo

Zdroj [2]

5. Možnosti řešení

Pro řešení zadané úlohy se nabízí několik možností z hlediska výběru vývojového prostředí a programovacího jazyka. V této kapitole se možností řešení budeme věnovat podrobněji a popíšeme si jejich výhody a nevýhody.

5.1 Programovací jazyk C++ a knihovna OpenCV

OpenCV (Open Source Computer Vision Library), jak už název napovídá, jedná se o volně šiřitelnou knihovnu. Knihovna poskytuje funkce a algoritmy pro počítačové vidění a zpracování obrazu. Knihovna obsahuje více než 2500 optimalizovaných algoritmů, které mohou být použity například pro detekci obličeje, rozpoznání objektů, sledování pohybu a další. Knihovnu je možné použít v prostředí jazyků C++, C, Python, Java a MATLAB. [3]

Instalace knihovny OpenCV je poměrně komplikovaná. Uživatel si musí stáhnout a nainstalovat několik programů, které bude k instalaci potřebovat a dodržet postup, který je uvedený v dokumentaci na stránkách docs.opencv.org.

Při instalaci jsem narazil na několik problémů, které se mi po přečtení internetových diskusí povedlo úspěšně odstranit. S knihovnou jsem pracoval ve vývojovém prostředí Microsoft Visual Studio 2010. Při vytvoření nového projektu bylo zapotřebí nalinkovat potřebné knihovny v konfiguraci projektu a nastavit potřebné závislosti. Program by jinak nebyl schopný s knihovnou pracovat.

Bakalářskou práci jsem začal řešit tímto způsobem, ale programování v C++ s využitím knihovny OpenCV mi přišlo zbytečně složité. Řešil jsem neustále problémy, které se týkaly omezení programovacího jazyka a správného použití funkcí z knihovny. Proto jsem se nemohl soustředit na samotnou podstatu bakalářské práce. Navíc se po seznámení s prostředím domnívám, že pomocí těchto nástrojů není možné dosáhnout uspokojivého výsledku mé aplikace.

5.2 Qt Creator

Qt Creator je multiplatformní vývojové prostředí pracující s frameworkem Qt. Framework Qt nabízí nástroje pro vytváření aplikací s přívětivým uživatelským rozhraním pro stolní počítače, embedded systémy a mobilní platformy. Framework umožňuje programátorovi zacílit aplikace na různé operační systémy stolních počítačů, embedded a mobilních zařízení s pouze jedním kódem. Modulární knihovny

tříd C++ a vývojářské nástroje umožňují vývojářům snadno vytvářet aplikace pro jeden operační systém, sestavit je a spustit na jiném operačním systému nebo architektuře procesoru. To vede k úspoře času i zvýšení efektivity programování aplikace. [4]

Instalace aplikace Qt Creator je jednodušší než u předchozího OpenCV. Instalace je k dispozici ke stažení v jednom balíčku. Ve vývojovém prostředí Qt Creator se pracuje velice pohodlně, rozhraní se ovládá intuitivně. Na internetu je k dispozici rozsáhlá dokumentace funkcí vývojového prostředí Qt Creator. Programování probíhá v jazyce C++, který využívá všechny funkce Qt knihoven. Vývojové prostředí obsahuje také designer pro návrh uživatelského rozhraní a nástroje pro debuggování.

5.3 National Instruments Vision Assistant

National Instruments (NI) Vision Assistant je nástroj pro vytváření a testování aplikací pro zpracování obrazu. Nástroj slouží pro vytvoření vlastního algoritmu z nabízených funkcí. Skriptovací funkce tohoto nástroje umožňuje sledovat každý krok algoritmu a výstup jednotlivých funkcí. Díky tomu se snadno provádí optimalizace algoritmu a odhalují chyby. Výsledný algoritmus lze otestovat na dalších vstupních datech a vyhodnotit jeho spolehlivost. Použitím LabVIEW VI Creation Wizard je možné z algoritmu vytvořit podprogram pro vývojové prostředí LabVIEW, zde ho dále upravovat a rozšiřovat. [9]

Nástroj NI Vision Assistant nabízí pro zpracování obrazu celou řadu funkcí, které jsou rozděleny do těchto skupin:

- Image Analysis Functions – Funkce pro analýzu obrazu;
- Color Image Processing Functions – Funkce pro zpracování barevných obrazů;
- Grayscale Image Processing Functions – Funkce pro zpracování šedotónových obrazů;
- Binary Processing Functions – Funkce pro zpracování binárních obrazů;
- Machine Vision Functions – Funkce strojového vidění.

5.4 Vývojové prostředí National Instruments LabVIEW

LabVIEW je vývojové prostředí pro grafický jazyk G, které vytváří společnost National Instruments. Používá se k programování především měřicích, řídicích a testovacích systémů. Jedná se o nástroj tzv. virtuální instrumentace, to znamená, že vývojové prostředí umožňuje vytvářet virtuální (tedy zdánlivé) programové řešení, kterým nahrazuje skutečné měřicí a indikační přístroje.

Program, se kterým se v LabVIEW pracuje, se nazývá virtuální přístroj - anglicky Virtual Instrument a odtud zkratka VI. Každé VI se skládá ze dvou částí:

- uživatelského rozhraní neboli čelního panelu, jak je v LabVIEW označováno;

- blokového diagramu, který obsahuje zdrojový kód VI v grafickém jazyce.

5.4.1 Čelní panel

Čelní panel vytváří uživatelské rozhraní pro řízení aplikace, které je tvořeno ovládacími a zobrazovacími prvky. Mezi ovládací prvky patří všechny vstupní zařízení, které ovlivňují běh programu. Jedná se o různé přepínače, tlačítka, klávesnice a pole pro zadávání hodnot. Tyto prvky jsou v LabVIEW označeny jako tzv. Controls. Mezi zobrazovací prvky patří diody pro zobrazení bitové informace, různé podoby číselných indikátorů, a další speciální indikátory. Tyto prvky označujeme jako tzv. Indicators. Mají na starosti poskytnutí informací z programu a jejich přenos na čelní panel.

5.4.2 Blokový diagram

Druhou částí každého VI je blokový diagram, v němž uživatel sestavuje vlastní algoritmus programu. Skládá se z jednotlivých prvků, které nám nabízí paleta Functions. Do blokového diagramu můžeme také přidat vlastní SubVI nebo-li podprogram, který bude řešit nějakou dílčí úlohu aplikace. Používání SubVI vede ke zpřehlednění blokového diagramu a jednoduššímu testování, protože můžeme postupně odladit jednotlivé části programu. Další výhodou SubVI je snadná znovupoužitelnost kódu v jiné aplikaci.

Blokový diagram je tvořený prvky vykonávajícími určitou funkci. K jejich propojení se používají spoje, které se dají přirovnat k proměnným v běžném programovacím jazyce. Spoje mohou být různého datového typu a tomu odpovídá jejich barva. [13, s. 18-33]

6. Návrh řešení

Po seznámení se s aplikací NI Vision Assistant jsem se rozhodl pro vývoj aplikace pro testování DPS pomocí tohoto nástroje. Zaujalo mě hlavně, že aplikace je přímo určena pro zpracování a vyhodnocení obrazových informací. V aplikaci NI Vision Assistant vytvořím skript, který zpracuje a vyhodnotí vstupní data. Ze skriptu se vygeneruje tzv. VI (podprogram), který se dále optimalizuje ve vývojovém prostředí LabVIEW 2012. V LabVIEW se také vytvoří uživatelské rozhraní aplikace.

6.1 Skript pro zpracování obrazu

Na začátku skriptu se provede načtení vstupního obrázku – naskenované DPS ve formátu TIFF. Následně funkce *Geometry* umožňuje jeho zrcadlení podle horizontální nebo vertikální osy. Tato operace je důležitá, protože je nutné, aby si obrázek předlohy a naskenovaný motiv odpovídaly. V případě, že by obrázky vůči sobě byly zrcadlově převrácené, nedošlo by k jejich porovnání.

Poté dojde k převodu obrazu do odstínů šedi pomocí funkce *Color Plane Extraction*, která pracuje v režimu *HSV – Value Plane*. To znamená, že z obrázku extrahuje hodnotu jasu barvy. Výsledkem je šedotónový obraz se zvýšeným kontrastem, což je pro další zpracování ideální.

Následně se provádí prahování pomocí funkce *Threshold*. Rozsah prahování je nastavený na 0 až 137. To znamená, že jsou z obrazu vybrány všechny pixely s touto hodnotou a všechny zbylé jsou nastaveny jako pozadí. Na výstupu této funkce je tedy binární obraz, ve kterém jedničky představují vyleptaný motiv na DPS – místa bez měděné vrstvy. Nuly naopak reprezentují pozadí a vodivé spoje – měděnou vrstvu DPS.

Po prahování musí být proveden převod zpět na šedotónový obraz pomocí funkce *Lookup Table* v režimu *Equalize*. Následující funkce *Geometric Matching* totiž nedokáže pracovat s binárním obrazem.

Funkce *Geometric Matching* vyhledá v obrázku motiv předlohy, který je jí předán v parametrech jako šablona. Vytvoření šablony není tak úplně intuitivní, jak by uživatel předpokládal. Proto jsem si pro vytvoření šablon vytvořil samostatný skript *CreateTemplate.vscr*. Obrázek předlohy ze softwaru pro návrh plošných spojů používá barevný model RGB a barevnou hloubku 8 bitů na kanál. Pro vytvoření šablony je třeba převést obrázek na binární 8-bit formát. K tomu slouží funkce *Color Plane Extraction* v režimu *RGB – Red Plane*, která z obrázku vybere pouze barevnou složku Red. Nezáleží na tom, jaká barevná složka je z obrázku vyextrahována,

protože všechny barevné složky (Red, Green, Blue) mají stejnou hodnotu. Následně je pomocí funkce *Geometric Matching* vytvořena nová šablona, kterou tato funkce vyhledává na naskenovaném obrázku a vrací souřadnice nalezených shod.

Z výsledků funkce *Geometric Matching* je nutné vytvořit systém souřadnic pomocí funkce *Set Coordinate System* v režimu Horizontal, Vertical and Angular Motion. Souřadnice X a Y, případně také úhlová odchylka - natočení, jsou předány další funkci *Golden Template Comparison*. Tato funkce provádí již samotné porovnání šablony a naskenovaného motivu. Výsledkem jsou nalezené rozdíly mezi těmito dvěma obrázky.

Při testování tohoto skriptu bylo zjištěno, že se u funkce *Golden Template Comparison* objevuje problém se zarovnáním podle souřadnic z funkce *Set Coordinate System*. V parametru Alignment se vyplňovaly jiné hodnoty souřadnic X a Y, než které byly funkci předány. V rámci testování byly souřadnice vyplněny ručně a řešení tohoto problému bylo odloženo do vývojového prostředí LabVIEW. K vytvoření VI pro LabVIEW byl použitý nástroj LabVIEW VI Creation Wizard, který je součástí programu Vision Assistant.

6.2 Geometric Matching

Vyhledávací a porovnávací algoritmy hledají oblasti v kontrolovaném obrázku, které obsahují podobné informace jako obrázek šablony. Tyto informace jsou nejprve zpracovány a poté z nich jsou vytvořeny charakterizující vlastnosti, které používá algoritmus pro nalezení shody. Algoritmus Geometric Matching primárně používá pro porovnání geometrické informace obsažené v obrázku šablony. Geometrické informace jsou hrany, křivky, případně složitější geometrické tvary tvořené křivkami.

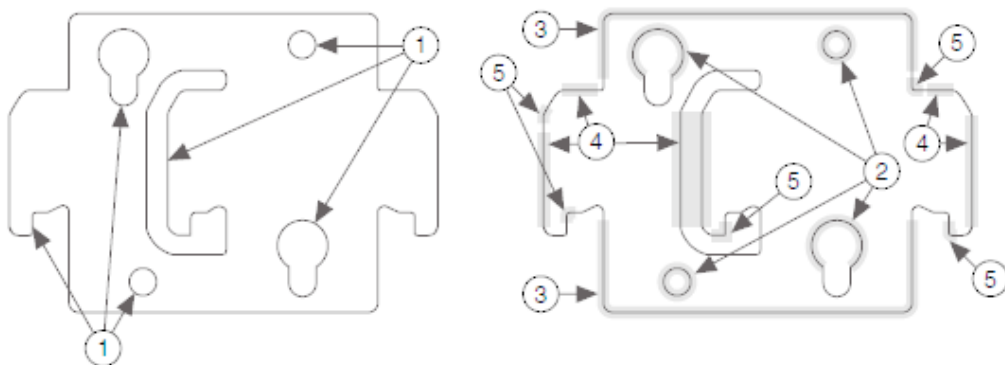
Proces geometrického porovnání se skládá ze dvou částí: učení a hledání shody. V procesu učení jsou z obrázku šablony extrahovány geometrické informace. Algoritmus poté uloží nalezené informace o tvarech a jejich vzájemných prostorových vlastnostech. Uložení těchto informací je provedeno takovým způsobem, aby bylo umožněno co nejrychlejší porovnání s kontrolovaným obrázkem. V nástrojích NI Vision se tyto informace ukládají do obrázku šablony. V průběhu hledání shody jsou extrahovány geometrické informace také z inspekčního obrázku. Poté jsou tyto informace porovnány s informacemi získanými z obrázku šablony a hledají se shody.

NI Vision nabízí dvě metody pro geometrické porovnání. Obě metody pracují s křivkami získanými z obrázků. Liší se tím, jakým způsobem jsou získané křivky použité pro porovnání. První metoda je založená na hranách (Edge-based) a vypočítává hodnotu gradientu hrany v každém bodě podél křivek. Pro porovnání používá metoda tuto hodnotu společně s informací o poloze bodu od středu šablony. Druhá metoda Feature-based extrahuje z křivek informace o geometrických tvarech a ty následně používá pro porovnání.

Obrázek 6.1 znázorňuje, jaké informace mohou být nalezeny v obrázku šablony a následně použity pro porovnání. V levé části je motiv popsán pomocí křivek, které odpovídají hranám v obraze. Tyto informace používá metoda Edge-based. V pravé části je vidět stejný motiv popsán podrobněji pomocí několika geometrických tvarů,

které používá metoda Feature-based. Geometrické tvary mohou být následující:

- 1 – křivky;
- 2 – kruhové tvary;
- 3 – obdélníkové tvary;
- 4 – přímky;
- 5 – rohy.



Obrázek 6.1: Geometrické tvary

Zdroj [10]

Metoda Edge-based funguje na libovolném tvaru, pokud je zaručeno, že se významná část motivu shoduje s motivem šablony. Metoda Feature-based funguje za předpokladu, že se motiv skládá z geometrických tvarů (např. kruhů, obdélníků a přímek).

6.2.1 Hledání křivek v obraze

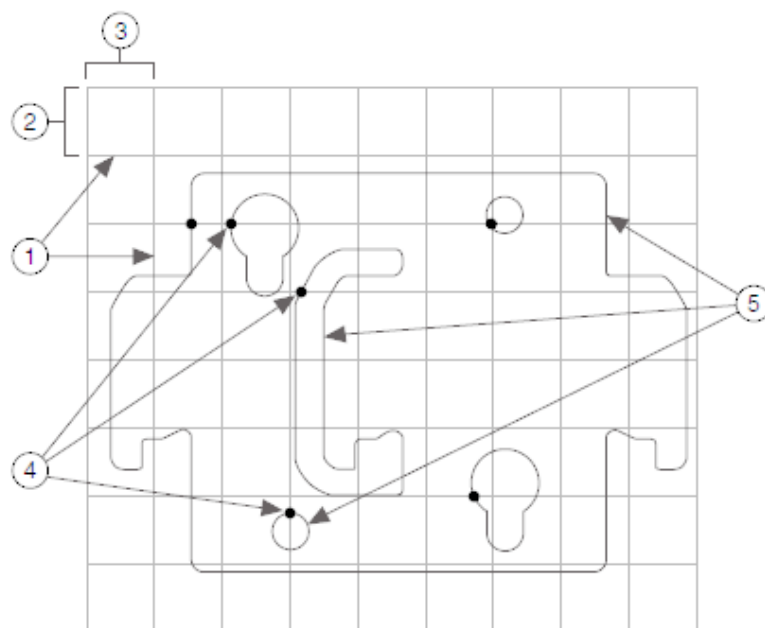
Křivka je v našem případě množina hranových bodů, které jsou propojeny a tvoří souvislý tvar. Nalezené křivky tedy tvoří obrysy jednotlivých částí v obraze. Hledání křivek se skládá ze tří kroků: nalezení prvního bodu, trasování křivky a závěrečného vylepšení.

Trasování křivky začíná v bodě, který má hranový kontrast větší než je nastavená prahová hodnota. Zároveň tento bod nemůže být již součástí nějaké jiné křivky. Kontrast hrany bodu se vypočítává pomocí hodnot jasů bodů, se kterými sousedí. Pro bod $P_{(i,j)}$ se hodnota kontrastu hrany vypočítá tímto způsobem:

$$\sqrt{(P_{(i-1,j)} - P_{(i+1,j)})^2 + (P_{(i,j-1)} - P_{(i,j+1)})^2}$$

Pro 8bit obrázek nabývá kontrast hrany hodnot od 0 do 360.

Procházení každého bodu v obraze by trvalo příliš dlouho, a proto algoritmus může procházet pouze některé řádky a sloupce. Velikost kroků se nastavuje parametry Row Step a Column Step. Pokud parametr Row Step bude nastavený například na hodnotu 5, bude algoritmus procházet každý 5. řádek. Algoritmus prochází nejprve řádky a poté sloupce. Trasování křivky začíná, když algoritmus nalezne počáteční hranový bod. Bod je přidán ke křivce, pokud má v okolí posledního bodu křivky největší kontrast hrany. Zároveň musí být tento kontrast větší než nastavená prahová hodnota. Tímto postupem jsou ke křivce přidávány další body. Ve chvíli, kdy již žádné body nelze přidat, se algoritmus vrátí do nalezeného počátečního bodu a opakuje hledání dalších bodů opačným směrem. Postup hledání a trasování křivek znázorňuje obrázek 6.2.



Obrázek 6.2: Hledání křivek

Zdroj [10]

Popis obrázku:

- 1 – rastr, na kterém jsou vyhledávány počáteční body křivek;
- 2 – hodnota parametru Row Step;
- 3 – hodnota parametru Column Step;
- 4 – nalezené počáteční body křivek;
- 5 – nalezené křivky.

Na závěr algoritmus provede následující vylepšení křivek:

- Dvě křivky jsou sloučeny, pokud jsou jejich koncové body blízko sebe.
- Křivka je uzavřena, pokud její koncové body jsou ve vzdálenosti nastavené parametrem Max Endpoint Gap.
- Křivka je odstraněna, pokud je kratší než vzdálenost nastavená parametrem Min Length.

[10]

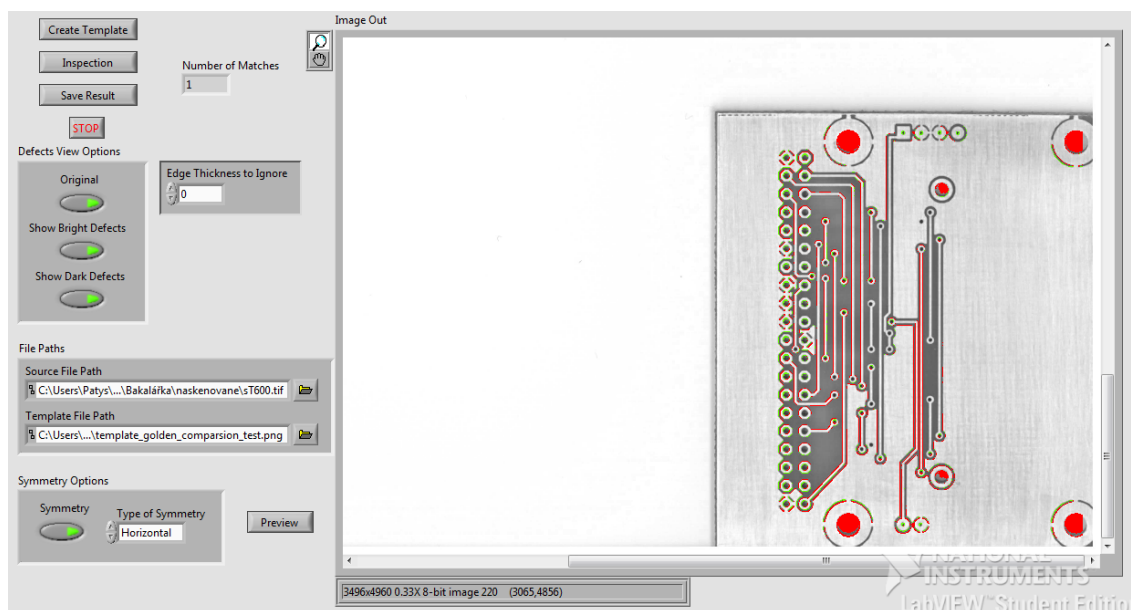
7. Aplikace pro testování DPS

Tato kapitola popisuje vytvoření konečné aplikace pro testování DPS ve vývojovém prostředí LabVIEW. Pro programování byl vytvořen projekt s názvem *PCB-Testing.lvproj*, který obsahuje několik podprogramů SubVI. Každé SubVI řeší část výsledné aplikace. Jejich funkce a způsob realizace jsou podrobně vysvětleny v následujícím textu.

7.1 Uživatelské rozhraní

Čelní panel VI s názvem *Main.vi* poskytuje uživatelské rozhraní pro ovládání aplikace. Nachází se zde tyto ovládací a zobrazovací prvky:

- **Create Template** – tlačítko pro vytvoření šablony, která je následně použita pro porovnání obrázků;
- **Inspection** – tlačítko pro provedení inspekce;
- **Save Result** – tlačítko pro uložení výsledku inspekce;
- **STOP** – tlačítko pro zastavení programu;
- **Matching** – indikátor informující o nalezené shodě motivu šablony na naskenovaném obrázku;
- **Defect View Options** – ovládací prvky umožňující uživateli nastavit jaké typy defektů chce zobrazit, a jestli mají být zobrazeny na naskenovaném motivu;
- **Edge Thickness to Ignore** – pole pro nastavení tloušťky hrany, která má být ignorována;
- **File Paths** – pole pro zadání cesty k umístění souborů pro porovnání;
- **Symmetry Options** – ovládací prvky pro nastavení zrcadlového převrácení naskenovaného obrázku podle horizontální nebo vertikální osy;
- **Preview** – tlačítko pro náhled nastavení symetrie;
- **Image Out** – indikátor zobrazující výstup po provedené inspekci.



Obrázek 7.1: Uživatelské rozhraní aplikace

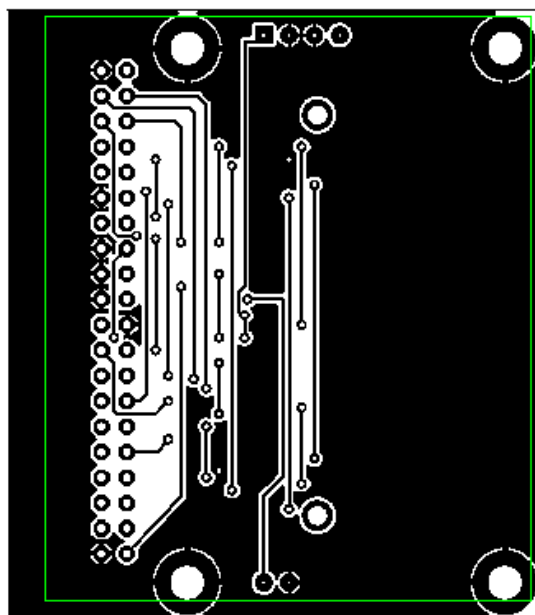
7.2 Vytvoření šablony

Kliknutím uživatele na tlačítko *Create Template*, se spustí SubVI s názvem *Create-Template.vi*. Toto SubVI vytváří šablonu pro funkce *Geometric Matching* a *Golden Template Comparison*. Původní řešení vytvořilo celkem dvě šablony - pro každou funkci jednu. Nakonec bylo upraveno a potřebná data se podařilo uložit pouze do jednoho souboru šablony, který následně používají obě funkce. Po spuštění SubVI se uživateli otevře okno aplikace, ve kterém zadá cestu k umístění souboru předlohy – *Pattern File Path*. Po zadání cesty se automaticky zobrazí náhled obrázku v indikátoru *Pattern Image*. Stisknutím levého tlačítka myši a následním tažením uživatel vybere oblast předlohy, kterou chce porovnat s naskenovaným motivem. Ideální je vybrat vnitřní část DPS bez okrajů viz obrázek 7.2. Výběr této oblasti je důležitý pro správné nalezení motivu pomocí funkce *Geometric Matching*.

Template Destination File Path je cesta, kam má být uložen soubor šablony. Šablona ve formátu PNG se vytvoří po kliknutí na tlačítko *Create* a následně se toto okno aplikace samo zavře. Cesta k šabloně se přenesení do hlavního okna aplikace a je automaticky vyplněna do odpovídajícího pole ve *File Paths*. Postup vytvoření šablony znázorňuje obrázek 7.3.

7.2.1 Šablona pro Geometric Matching

Šablona je vytvořena z oblasti, kterou vybral uživatel na obrázku předlohy. Oříznutí zajišťuje funkce *IMAQ Extract* (1), které je předána vybraná oblast jako parametr ROI (Region of Interest) z indikátoru *Pattern Image*. Následující funkce *IMAQ Learn Geometric Pattern 2* (3) přidává popis k obrázku šablony. Tyto data používá funkce *Geometric Matching* při hledání motivu na naskenovaném obrázku. Data nastavení



Obrázek 7.2: Výběr oblasti pro porovnání

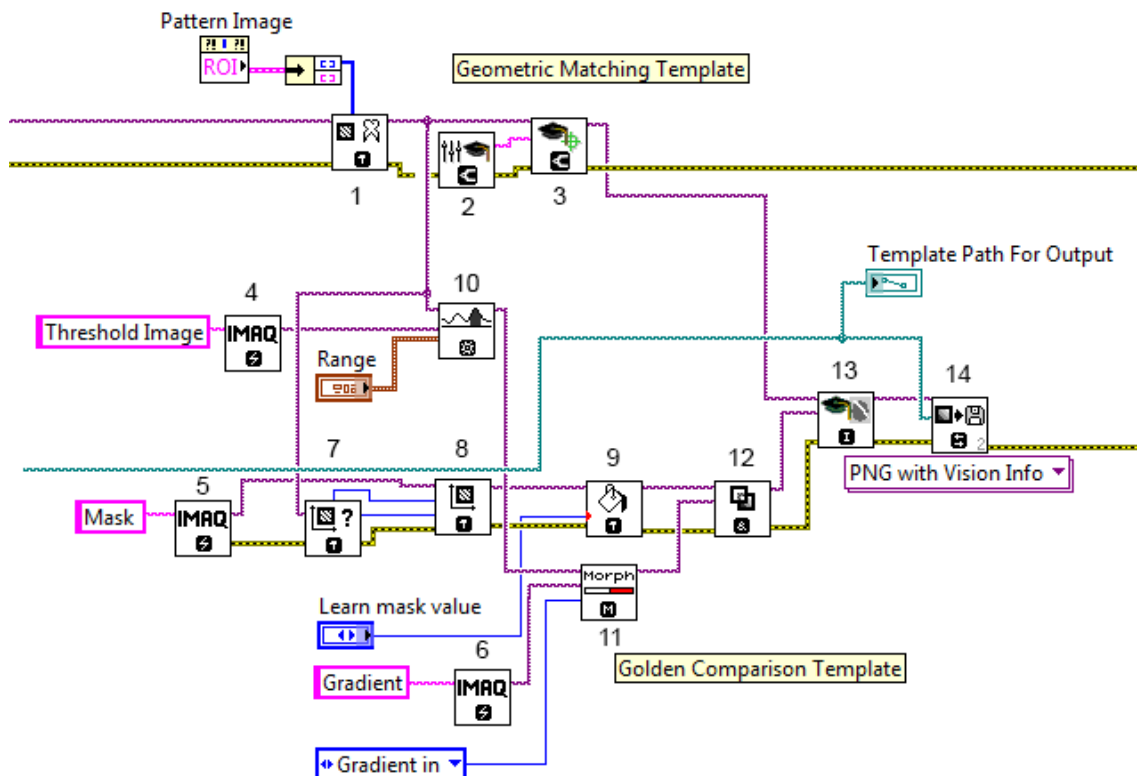
jsou předána pomocí funkce *IMAQ Setup Learn Geometric Pattern 2* (2). V této funkci jsou nastaveny parametry křivek (Curve Parameters) na výchozí hodnoty. Obrazová data šablony včetně dat nastavení jsou předána funkci *IMAQ Learn Golden Template* (13), která vytváří šablonu pro funkci *Golden Template Comparison*.

7.2.2 Šablona pro Golden Template Comparison

Při vytváření této šablony je možné připojit také tzv. masku. Maska obsahuje hrany, které mohou být ignorovány při porovnávání. Šířka ignorované hrany se nastavuje pomocí parametru Edge Thickness To Ignore v uživatelském rozhraní aplikace.

Před vytvořením šablony se nejprve alokuje paměť pro výsledky z jednotlivých funkcí. K alokaci paměti se používá funkce *IMAQ Create* (4, 5 a 6). Funkce *IMAQ GetImageSize* (7) zjišťuje rozměry uživatelem vybrané části předlohy. Maska musí být stejných rozměrů, a proto je její rozlišení nastaveno na stejné hodnoty pomocí funkce *IMAQ SetImageSize* (8). Celý obrázek masky je následně vyplněn funkcí *IMAQ FillImage* (9) hodnotou 1, která reprezentuje hranu, která může být ignorována. K tomuto nastavení je k funkci připojen výčtový typ Learn Mask Value, který obsahuje hodnoty pixelů, jakých může nabývat obrázek masky. Tyto hodnoty jsou následující:

- 0 – Default Behavior – výchozí chování;
- 1 – Ignored Edge – odpovídající pixel tvoří hranu a může na něj být aplikován parametr Edge Thickness To Ignore;
- 2 – Ignored – odpovídající pixel je při porovnání ignorován.

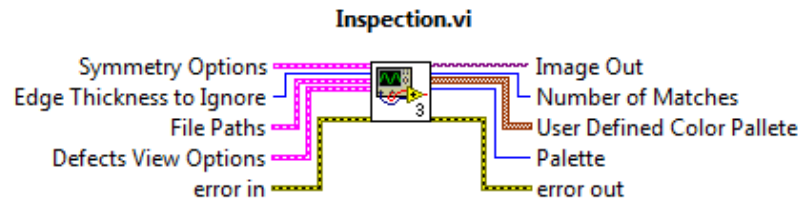


Obrázek 7.3: Vytvoření šablony

Funkce *Threshold* (10) provede prahování oříznutého obrázku předlohy s rozsahem nastaveným na 150 až 255. Díky tomu vznikne binární obrázek, na kterém jsou všechny pixely s hodnotou v tomto rozsahu reprezentovány hodnotou 1. Jedná se o všechna světlá místa na obrázku předlohy. Pixely s hodnotou mimo rozsah reprezentuje hodnota 0. S výstupem prahování pracuje funkce *IMAQ Morphology* (11) v režimu *Gradient in*. Funkce z objektů na obrázku vyextrahuje pouze jejich vnitřní hrany. Hrany jsou reprezentovány hodnotou 1, stejně jako ve vstupním binárním obrázku. Získané hrany jsou přivedeny na vstup funkce *IMAQ Mask* (12), která pracuje také s výstupem funkce *IMAQ FillImage*. Funkce *IMAQ Mask* ponechá hodnotu nastavenou funkcí *IMAQ FillImage* v místech, kde jsou hrany. Ostatní místa jsou nastavena na 0. Vzniklou masku a šablonu pro funkci *Geometric Matching* použije funkce *IMAQ Learn Golden Template* (13). Výslednou šablonu uloží funkce *IMAQ Write File 2* (14) ve formátu PNG with Vision Info.

7.3 Inspekce

Inspekci uživatel zahájí kliknutím na tlačítko *Inspection*, které spustí SubVI s názvem *Inspection.vi*. Schéma SubVI znázorňuje obrázek 7.4. V levé části jsou zobrazené vstupy z hlavního VI *Main.vi*. V pravé části jsou výstupy, které toto SubVI poskytuje.



Obrázek 7.4: Schéma Inspection.vi

7.3.1 Příprava pro porovnání

Na začátku tohoto SubVI dochází k načtení vstupních souborů pro porovnání. Cesty k umístění těchto souborů jsou předány pomocí datové struktury cluster s názvem *File Paths*. Cluster může seskupovat několik vstupních či výstupních prvků a pomáhá zredukovat počet datových spojů. Cluster totiž spojuje jenom jeden datový spoj, což vede ke zpřehlednění programu. Na začátku SubVI jsou z clusteru získány jednotlivé cesty k souborům pomocí funkce *Unbundle By Name*. Po alokaci paměti funkcí *IMAQ Create* je vstupní soubor ze scanneru načtený funkcí *IMAQ ReadFile*. Vstupní soubor šablony načítá funkce *IMAQ Read Image And Vision Info*, která navíc zpracuje i data, která byla k šabloně přidána při jejím vytváření (viz kapitola 7.2).

Druhý cluster *Symmetry Options* obsahuje informaci o tom, zde je vyžadováno zrcadlové převrácení naskenovaného obrázku či nikoliv – parametr *Symmetry*. Případně také informaci, jestli má být obrázek převrácen podle vodorovné či svislé osy – parametr *Type Of Symmetry*. Požadované převrácení obrázku provádí funkce *IMAQ Symmetry* připojená k tomuto clusteru.

Funkce *IMAQ ExtractSingleColorPlane* z naskenovaného obrázku extrahuje hodnotu jasu barvy. Vznikne tak 8bitový šedotónový obraz se zvýšeným kontrastem, ve kterém se vyskytují barvy v rozsahu 0 až 255. Nejtmavší černá barva odpovídá hodnotě nula a nejsvětlejší bílá hodnotě 255. Následuje funkce *IMAQ Threshold*, která provede prahování v rozsahu 0 až 137. Funkce nahradí pixely s hodnotou v tomto rozsahu hodnotou 1 a všechny zbylé hodnotou 0. Vznikne tedy binární obrázek, ve kterém jsou místa bez měděné vrstvy reprezentována hodnotou 1 a měděné spoje hodnotou 0. Pro správné fungování následujících funkcí je zapotřebí převést obraz zpět na 8bitový, ve kterém budou obsaženy pouze dvě hodnoty - 0 a 255. K převodu slouží funkce *IMAQ Equalize*. S takto zpracovaným obrázkem ze skeneru pracuje funkce *IMAQ Match Geometric Pattern 2* a *IMAQ Compare Golden Template*, které jsou popsány níže.

7.3.2 Hledání motivu

Funkce *IMAQ Match Geometric Pattern 2* v naskenovaném obrázku hledá odpovídající motiv šablony. Princip funkce popisuje kapitola 6.2. Funkci předává parametry funkce *IMAQ Setup Match Geometric Pattern 2*, ke které je připojeno několik clusterů s nastavením. Cluster *Curve Params* obsahuje nastavení pro vyhledávání křivek v naskenovaném obrázku. Nastavení je ponecháno stejně jako při vytváření

šablony na výchozí hodnoty. Cluster Match Mode slouží k nastavení podmínek pro hledání motivu. Podmínky mohou být následující:

- Rotation – motiv může být natočený;
- Scale – motiv může mít jiné měřítko;
- Occlusion – jaká část motivu může být překryta.

Minimální a maximální hodnota rotace, měřítka a okluze se nastavuje pomocí pole Range Settings. Rozsah pro natočení je nastavený na 0-360°.

K funkci *IMAQ Match Geometric Pattern 2* jsou samostatně připojeny ještě dva parametry – Number of Matches Requested a Minimum Match Score. Prvním parametrem se udává, kolik shod má být v obrázku vyhledáno. Tento parametr je nastavený na hodnotu 1. Druhým parametrem je nastaveno minimální skóre pro nalezený motiv. Skóre nabývá hodnot v rozsahu 0 až 1000. Čím větší je dosažené skóre, tím více se shoduje obrázek šablony s naskenovaným motivem. Minimální skóre je nastaveno na hodnotu 800. Nalezené shody, které mají skóre menší, budou ignorovány.

Výsledky z funkce *IMAQ Match Geometric Pattern 2* jsou reprezentovány polem clusterů. Pro přístup k výsledkům je použita funkce *Index Array*, která vrátí první cluster. Z tohoto clusteru jsou poté získány informace o souřadnicích, úhlu natočení a měřítku, ze kterých je vytvořený další cluster Alignment. Tento cluster je důležitý pro funkci *IMAQ Compare Golden Template*, pro kterou určuje zarovnání šablony.

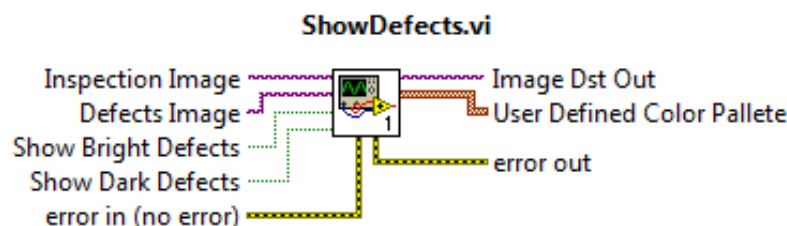
7.3.3 Porovnání motivu

Funkce *IMAQ Compare Golden Template* porovnává intenzitu jednotlivých pixelů obrázku šablony a naskenovaného obrázku. Pixel na naskenovaném obrázku je označený jako defekt v případě, že se liší od odpovídajícího pixelu na obrázku šablony. Nastavení této funkce se provádí clusterem Inspection Options, který obsahuje několik parametrů. Nejdůležitějším parametrem je Edge Thickness To Ignore, který určuje, jaká šířka hrany bude ignorována. Tento parametr nastavuje přímo uživatel z čelního panelu aplikace. Také je nastavený binární režim, což znamená, že nalezené defekty budou reprezentovány následujícím způsobem. Místa, ve kterých je kontrolovaný obrázek světlejší než obrázek šablony, mají hodnotu 2 a jsou zvýrazněna zeleně. Místa, ve kterých je kontrolovaný obrázek naopak tmavší než obrázek šablony, mají hodnotu 1 a jsou zvýrazněna červeně. Zbylé pixely mají hodnotu 0 – černá barva. Nalezené defekty funkce uloží do jednoho obrazu a poté se s nimi dále pracuje.

Obraz s defekty je přivedený do funkce *IMAQ RemoveParticle*, která má za úkol odstranit z něj menší defekty.

7.3.4 Zobrazení defektů

Nalezené defekty je třeba zobrazit na výstupu. To má za úkol SubVI *ShowDefects.vi*, jehož schéma znázorňuje obrázek 7.5.



Obrázek 7.5: Schéma ShowDefects.vi

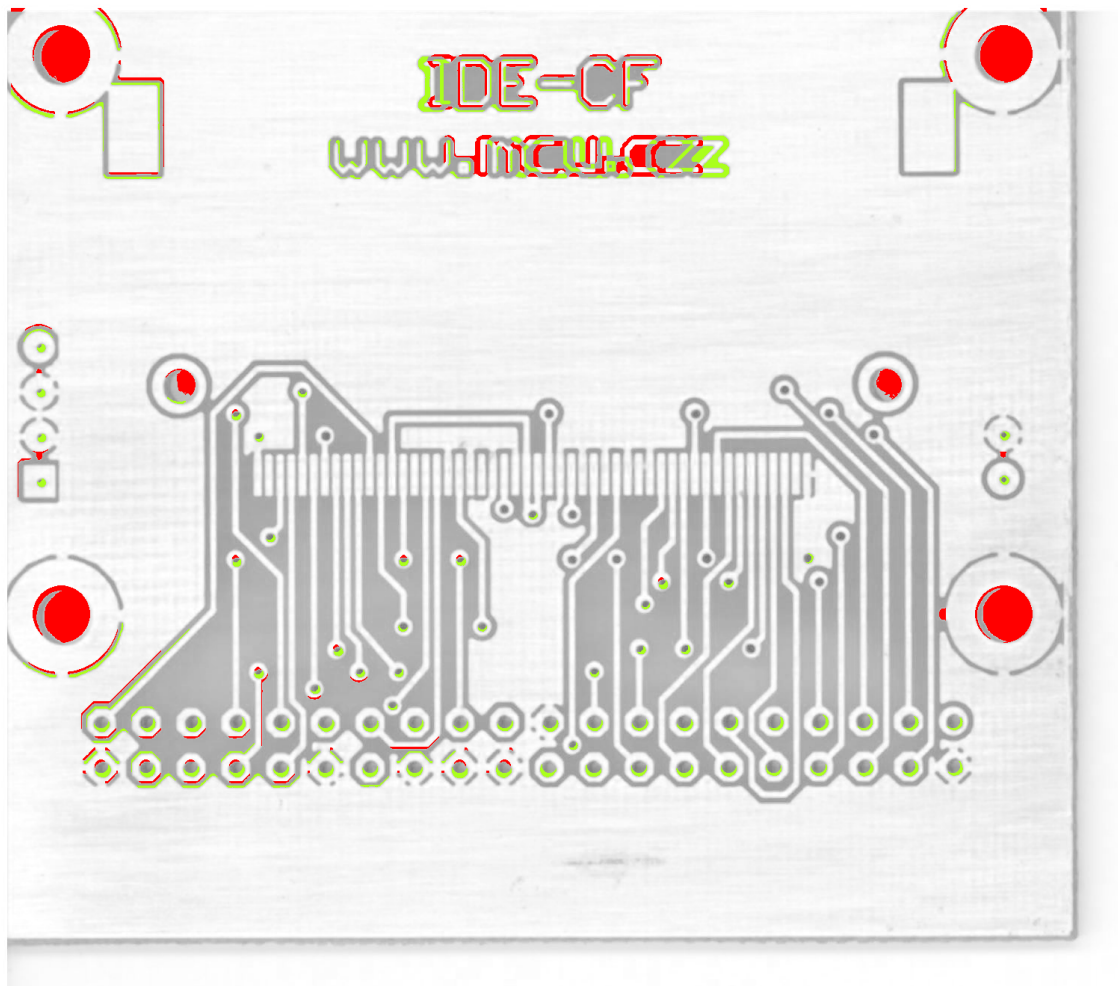
Z clusteru Defects View Options jsou předány informace o tom, jaké defekty chce uživatel zobrazit a zda mají být zobrazeny na naskenovaném motivu. V případě, že uživatel chce zobrazit pouze samotné defekty, je vytvořený nový obrázek, který má stejné rozměry jako ten naskenovaný. Prázdný obrázek je vyplněný šedou barvou funkcí *IMAQ FillImage*. Šedý odstín byl zvolený, aby světle zelené i červené defekty byly dobře viditelné. Jako vstup Inspection Image je tedy SubVI předaný buď prázdný šedý obrázek, nebo naskenovaný motiv převedený do odstínů šedi. SubVI vytváří vlastní barevnou paletu, která vychází z šedotónové palety. V šedotónové paletě se vyskytují hodnoty 0 až 255. V barevném modelu RGB mají tyto hodnoty stejné zastoupení ve všech barevných složkách. Např. hodnota 125 bude v RGB reprezentována hodnotou (125,125,125). SubVI upravuje zastoupení barevných složek pro hodnoty 1 a 2. Jak již bylo popsáno výše, hodnota 1 označuje tmavé defekty a hodnota 2 defekty světlé. Pro hodnotu 1 je hodnota RGB nastavena na (255,0,0) – červený odstín. Pro hodnotu 2 je hodnota RGB nastavena na (103,255,3) – světle zelený odstín. V případě, že uživatel nechce světlé nebo tmavé defekty zobrazit, není pro ně ani upravována barevná paleta.

Na naskenovaný obrázek je aplikována funkce *IMAQ Compare* s operátorem MAX. Funkce porovnává hodnoty jednotlivých pixelů s nastavenou konstantou - hodnotou 3. Hodnoty v naskenovaném obrázku, které jsou menší než 3, jsou přepsány na hodnotu 3. Znamená to, že v naskenovaném obrázku zůstanou hodnoty 3 až 255. Zároveň jsou také pixely s hodnotou 0 v obrázku s defekty nahrazeny hodnotou 255 pomocí funkce *IMAQ UserLookup*. Pokud uživatel nechce zobrazit světlé nebo tmavé defekty, jsou navíc hodnotou 255 nahrazeny i odpovídající hodnoty defektů. Nakonec se porovnává obrázek s defekty a naskenovaný obrázek pomocí funkce *IMAQ Compare* s operátorem MIN. Menší hodnota pixelů bude zachována na výstupu. Proto se hodnota 255 používá v obrázku s defekty pro pixely, které nemají být zobrazeny na výstupu.

7.3.5 Uložení výsledku

Výsledek inspekce si uživatel může uložit kliknutím na tlačítko Save Result, které se nachází na čelním panelu programu. Uložení se provádí v hlavním VI, kde je pomocí programové struktury *Event Structure* odchycena událost stisknutí tlačítka. Cestu pro uložení výsledku generuje SubVI *NewFileName.vi*, které jako zdroj používá cestu k naskenovanému obrázku. V tomto umístění je vytvořena nová složka Inspected,

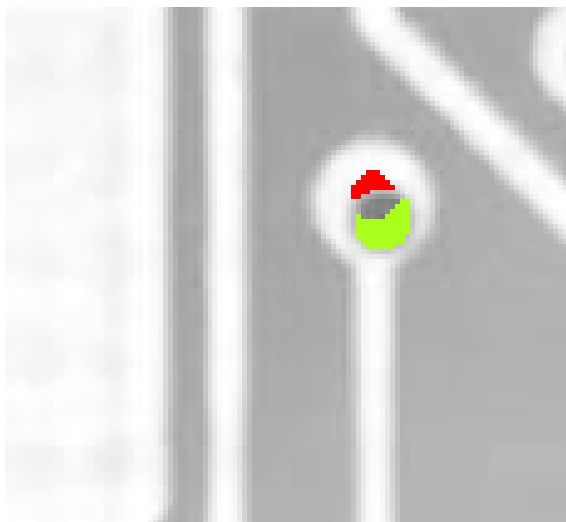
která bude obsahovat uložené výsledky inspekce. Název naskenovaného souboru bez přípony se vyjme pomocí funkce *Get File Extension* a přidá se k němu přípona .png. V případě, že uživatel ukládá více výsledků pro jeden kontrolovaný obrázek, je za jméno souboru přidáno ještě podtřítko a číslo. Vytvořená cesta k umístění souboru se předává funkci *IMAQ Write File 2*, která výsledek uloží ve formátu PNG. Funkce používá při uložení obrázku upravenou barevnou paletu, která byla definována v SubVI *ShowDefects.vi*. Výsledek inspekce znázorňuje obrázek 7.6.



Obrázek 7.6: Výsledek inspekce

8. Vyhodnocení aplikace

Aplikace byla otestována na dodaných vzorcích. Motiv na obrázku předlohy i motiv naskenované DPS byly dodány se stejnými rozměry. S těmito vstupními daty aplikace funguje spolehlivě a podle očekávání. Provedená inspekce dokáže odhalit posunutí vyvrtaných otvorů (viz obrázek 8.1) a odlišnosti tvaru motivu na vyrobené DPS. Je však třeba brát ohled na kvalitu naskenovaného obrázku, na kterém nejsou dokonale ostré hrany. Na hranách proto vznikají chybně detekované defekty. Tento problém je minimalizován tím, že uživatel může nastavit tloušťku hrany, která bude ignorována. Ideální hodnota tohoto parametru se u testovaných vzorků pohybovala kolem hodnoty 2 až 3.



Obrázek 8.1: Detail posunutí vyvrtaného otvoru

Dále se objevily defekty detekované uvnitř vyvrtaných otvorů v DPS. Tento problém u otvorů nastává, protože podle šablony by v místě otvoru měla být tmavá barva. Úhel osvětlení a snímané plochy způsobuje zkreslení, které nelze algoritmicky odstranit. Ve vyvrtaných otvorech se objevuje částečný stín a prosvítající světlé pozadí, které bylo za DPS při skenování. Tento problém byl vyřešen umístěním naopak tmavého pozadí za DPS při jejím skenování.

8.1 Metoda Feature-based

Původně se podle teorie jako nejlepší algoritmus pro vyhledání motivu zdál Geometric Matching s využitím metody Feature-based. Při použití této metody popsané v kapitole 6.2 bylo zjištěno, že motiv DPS je příliš složitý. Vyhledání geometrických tvarů a jejich vzájemným vlastností trvalo v obrázku předlohy i v kontrolovaném obrázku 3 minuty. Dohromady tedy vyhodnocení trvalo přibližně 6 minut, což je pro praktické využití nevyhovující. Testovaná DPS měla rozměry přibližně 7x6 cm. Při větších rozměrech DPS se dá předpokládat, že by zpracování trvalo ještě mnohem déle. Nutno podotknout, že metoda fungovala opravdu přesně, i pokud předloha měla jiné rozměry než kontrolovaný obrázek.

8.2 Metoda Edge-based

Při použití algoritmu Geometric Matching s metodou Edge-based, která je zaměřena pouze na hrany, byla inspekce provedena rychle. Při stejných rozměrech obrázku předlohy i kontrolovaného obrázku bylo dosaženo i potřebné přesnosti. Problémy se ovšem projevíly, pokud se rozměry předlohy změnily. Malá nepřesnost v určení polohy motivu na kontrolovaném obrázku vedla ke špatným výsledkům celé inspekce. Vůči výsledkům metody Feature-based byla souřadnice Y nalezené shody posunuta například o 7 pixelů. Nepřesně bylo také zjištěno měřítko nalezeného motivu vůči předloze. Přesná metoda Feature-based například zjistila, že měřítko odpovídá hodnotě 111,84 %. Metoda Edge-based vyhodnotila měřítko na 111,16 %. Rozdíl necelých 0,7 % má velký vliv na následné porovnání šablony metodou *Golden Template Comparison*.

Závěr

Cílem této práce bylo seznámení se s algoritmy pro zpracování obrazu a s vhodným vývojovým prostředím pro návrh aplikace. Získané znalosti poté využít při návrhu aplikace pro vyhodnocení obrazových dat. Aplikaci otestovat na dodaných vzorcích a vyhodnotit její možnosti. Tyto cíle byly naplněny, jak dokazuje aplikace pro testování DPS uložená na přiloženém CD.

Vývoj aplikace probíhal pomocí nástroje Vision Assistant a vývojového prostředí LabVIEW od firmy National Instruments. Pro správné fungování aplikace bylo nutné vyřešit i programové sestavení šablon, které se používají pro nalezení a porovnání motivu předlohy na naskenovaném obrázku DPS. Aplikace analyzuje defekty a poskytuje je na výstupu uživateli. Defekty mohou být zobrazeny samostatně nebo přímo na kontrolovaném obrázku DPS. Výsledek porovnání je možné uložit ve formátu PNG.

Tato metoda testování vyhodnotí místa možných defektů a proškolený uživatel by měl vyhodnit, zda nalezené místo defektu bude mít vliv na funkčnost DPS či nikoliv.

Pro nalezení motivu byl použit algoritmus Geometric Matching s využitím metody Edge-based, která na dodaných vzorcích fungovala spolehlivě a rychle. Metoda Feature-based se pro praktické použití neosvědčila, protože motiv DPS pro ni byl moc složitý a vyhodnocení trvalo v řádech minut.

Vlivem úhlu mezi osvětlením a snímanou plochou dochází ke zkreslení motivu uvnitř skenovaných otvorů. Algoritmicky je toto zkreslení neodstranitelné a pro jeho minimalizaci bylo nutno použít během skenování černé pozadí.

Velkým přínosem této práce je, že vytvořená aplikace by mohla nahradit drahé optické testery, které jsou pro inspekci DPS vyráběny.

Aplikace funguje pro obrázek předlohy a naskenovaný obrázek v měřítku 1:1. Naskýtá se možnost aplikaci rozšířit o kontrolu v případě jiného měřítka, kdy je jeden z obrázku menší/větší než ten druhý. V praxi by se tento typ problému neměl objevit, proto se jím tato práce nezabývala. Při běžném postupu software pro návrh plošných spojů vyexportuje motiv předlohy například v rozlišení 600 DPI a měřítku 1:1. Pokud bude i plochý scanner nastavený na rozlišení 600 DPI, tak si získané obrázky musí rozměrově odpovídat.

Literatura

- [1] Automatic Optical Inspection. *Orbotech* [online]. Orbotech, © 2014 [cit. 2014-01-15]. Dostupné z: http://www.orbotech.com/Eng/D2_ProductsGroup/MenuID/566/
- [2] Flying Probe Tester. *Atg Luther & Maelzer* [online]. Atg Luther & Maelzer, © 2013 [cit. 2014-01-15]. Dostupné z: http://www.atg-lm.com/atglm.nsf/html/products_A5_Neo
- [3] About. *OpenCV* [online]. Itseez, © 2014 [cit. 2014-02-04]. Dostupné z: <http://opencv.org/about.html>
- [4] Qt Creator. *Qt Project* [online]. Qt Project Hosting, © 2014 [cit. 2014-02-26]. Dostupné z: <https://qt-project.org/wiki/Category:Tools::QtCreator>
- [5] Clyde F. Coombs, J.: *Printed Circuits Handbook*. The McGraw-Hill Companies, 2008.
- [6] Jayaraman: *Digital Image Processing*. McGraw-Hill Education (India) Pvt Limited, 2011, ISBN 9780070144798.
- [7] Miano, J.: *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*. ACM Press Series, Addison Wesley, 1999, ISBN 9780201604436.
- [8] Mišurec, J.; Zeman, V.; Štěpán, M.: *Konstrukce elektronických zařízení - návrh plošných spojů*. 2003, skripta, Vysoké učení technické v Brně.
- [9] National Instruments: *NI Vision Assistant Tutorial*. 2003, [cit. 2014-02-26]. Dostupné z: <http://www.ni.com/pdf/manuals/322228d.pdf>
- [10] National Instruments: *Geometric Matching Technique*. 2011. Dostupné z: http://zone.ni.com/reference/en-XX/help/372916L-01/nivisionconcepts/geometric_matching_technique/
- [11] Plíva, Z.: *Z historie plošných spojů*. *Časopis DPS* [online], 1/2010, [cit. 2014-02-04]. Dostupné z: <http://www.dps-az.cz/media/pdf/historie/9-pliva-clanek.pdf>
- [12] Sayood, K.: *Lossless Compression Handbook*. Communications, Networking and Multimedia, Elsevier Science, 2002, ISBN 9780080510491.

- [13] Vlach, J.; Havlíček, J.; Vlach, M.: *Začínáme s LabVIEW*. Nakladatelství BEN, 2008.

A. Obsah přiloženého CD

- Složka "Aplikace" – instalátor aplikace pro testování DPS
- Složka "Obrázky" – obrázky použité pro testování a výsledky aplikace
- Složka "Zdrojové kódy" – zdrojové kódy aplikace
- Soubor "Bakalarska_prace_Odchazel_2014.pdf" – text bakalářské práce